Real-time speaker identification using the AEREAR2 event-based silicon cochlea

Li, C H ; Delbruck, T ; Liu, S C

Abstract: This paper reports a study on methods for real-time speaker identification using the output from an event-based silicon cochlea. These methods are evaluated based on the amount of computation that needs to be performed and the classification performance in a speaker identification task. It uses the binaural AEREAR2 silicon cochlea, with 64 frequency channels and 512 output neurons. Auditory features representing fading histograms of inter-spike intervals and channel activity distributions are extracted from the cochlea spikes. These feature vectors are then classified by a linear Support Vector Machine, which is trained against a subset of 40 speakers (20/20 male/female) from the TIMIT database. Speakers are correctly identified at >90% accuracy during each sentence utterance and with an average latency of 700±200ms from the start of the sentence.

DOI: https://doi.org/10.1109/ISCAS.2012.6271438

Originally published at:
Li, C H; Delbruck, T; Liu, S C (2012). Real-time speaker identification using the AEREAR2 event-based silicon cochlea. In: IEEE International Symposium on Circuits and Systems (ISCAS) 2012, Seoul, South Korea, 20 May 2012 - 23 May 2012, 1159-1162.
DOI: https://doi.org/10.1109/ISCAS.2012.6271438

Real-Time Speaker Identification using the AEREAR2 Event-Based Silicon Cochlea

Cheng-Han Li, Tobi Delbruck, and Shih-Chii Liu

Institute of Neuroinformatics, University of Zürich and ETH Zürich

Abstract—**This paper reports a study on methods for real-time speaker identification using the output from an event-based silicon cochlea. These methods are evaluated based on the amount of computation that needs to be performed and the classification performance in a speaker identification task. It uses the binaural AEREAR2 silicon cochlea, with 64 frequency channels and 512 output neurons. Auditory features representing fading histograms of inter-spike intervals and channel activity distributions are extracted from the cochlea spikes. These feature vectors are then classified by a linear Support Vector Machine, which is trained against a subset of 40 speakers (20/20 male/female) from the TIMIT database. Speakers are correctly identified at >90% accuracy during each sentence utterance and with an average latency of 700±200ms from the start of the sentence.**

Keywords: AER, spike-based, neuromorphic, cochlea, speaker identification, audition, real-time

I. INTRODUCTION

The auditory nerve output of biological cochleas consists of an asynchronous stream of spike events. Using a spike-event representation in models of auditory processing could lead to improved machine audition that is robust to noise and distractors. These models can offer further insight into human speech understanding under noisy conditions in comparison to machines [1].

Uysal and colleagues, for example, have used the timing information carried by the auditory nerve fiber spike trains in an Automatic Speech Recognition (**ASR**) task [2]. They used phase-synchrony coding as determined from the inter-spike time interval (**ISI**) histogram for each channel of a software model cochlea. They observe that the peaks of the ISI histogram across channels are aligned even with an extremely noisy vowel input signal. By exploiting the degree of phase synchrony features and feeding them to a Liquid State Machine for recognition, they found that the performance of this spike-based classifier on vowel phonemes from the TIMIT database was better in the presence of noise, particularly at low SNR levels down to 5dB, when compared to the performance of a conventional single-state HMM-based classifier using a 39-dimensional MFCC feature set with first and second derivatives.

Their work shows that the cues inherent in the auditory nerve fiber spike trains could be used as a competitive feature set for ASR and might provide one of the possible reasons for the superb robustness of the human auditory system.

The computational speed of these spike-based models can increase dramatically if implemented in hardware. Silicon cochlea chips provide real-time processing of the input sounds. Spike-based cochlea architectures transmit the cochlear outputs in an asynchronous way similar to the outputs of auditory nerve fibers [3][4][5][6][7][8]. This output representation encourages the use of the precise timing information carried by acoustic inputs. The latest spike-based cochlea system (AEREAR2) which is used in this work [8], combines features of previous cochlear designs that are robust to mismatch, with novel features for easier programmability of the architecture and operating parameters. The AEREAR2 gives us an opportunity to study the role of temporal information carried by the cochlea output events (or spikes) especially in real-time scenarios.

This paper describes event-driven methods to extract feature vectors from the asynchronous spikes of the AEREAR2 cochlea. Section II describes the AEREAR2 system. Section III describes several methods for determining when feature vectors should be extracted and how they are extracted, so that the system responds in real-time under normal speech conditions. Section IV describes the methods and results of a speaker identification task based on the considered methods for extracting features.

II. BACKGROUND

The AEREAR2 is a highly integrated spike-based silicon system built around a custom VLSI spike-based cochlea chip [8]. The cochlea has two matched 64-stage cascaded filter banks (Fig. 1), allowing connection to two electret microphones. It includes local adjustment and enhancement of filter sharpness, and has on-chip digitally controlled biases and microphone preamplifiers.

A prototype bus-powered USB board (Fig. 2) with

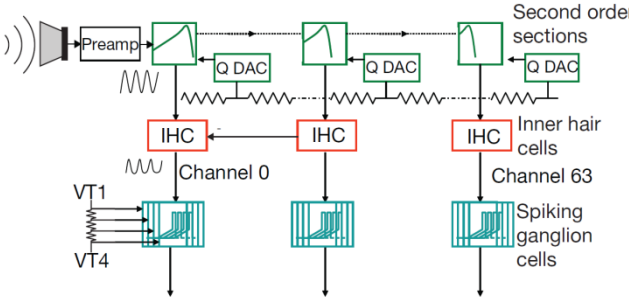

Fig. 1. Architecture of one side of the AEREAR2 binaural 64 channel cochlea with 4 neurons per channel. IHC: Inner Hair Cell. Q DAC=: Quality factor control. VT1-4: neuron thresholds.

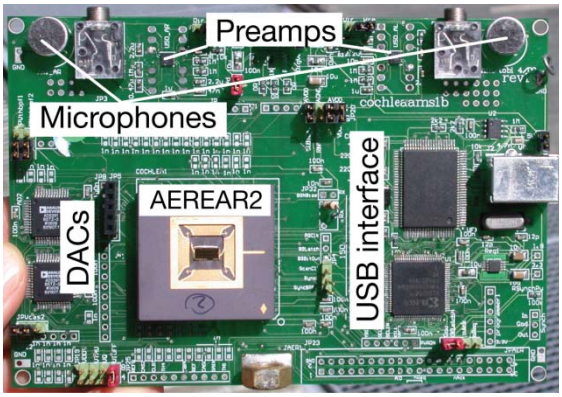

Fig. 2. AEREAR2 binaural 64 channel event-based cochlea system with a USB interface.

integrated microphones interfaces to jAER, an open-source software project for processing AER output [9]. The time-stamped events (with 1us resolution) are sent to a PC where they are further processed digitally for applications. On-chip MAX9814 microphone preamplifiers are used for natural sounds. For the measurements shown here, input was applied from a PC sound card to the cochlea for analysis of channel responses.

III. FEATURE EXTRACTION

In previous work, features extracted from the spike outputs from the binaural AEREAR2 system were used in a speaker identification task [10][11]. The results in [11] show comparable results between feature vectors extracted from the cochlea spikes and the MFCC features in a speaker identification task. However both the spike feature vectors and the MFCC features were extracted in regular fixed time window sizes. This processing within regular windows is computationally expensive in a system which should continuously identify speakers. This paper presents an investigation into alternate ways of extracting feature vectors only when necessary, with the aim of future use in power-constrained audition.

The spike features described in [10] are reconsidered here in this work. An example of the spike rasters from both the left and right cochlear channels in response to a sentence in the TIMIT database is shown in Fig. 1. Each of the 64 cochlear channels of each ear has four neurons. These neurons have been adjusted here to have nominally-identical firing thresholds. The spike rasters have been corrected for the increasing delay in the output spikes of the different channels because of the cascaded second-order section architecture of the system. The recorded timestamps in response to an impulse are subtracted from the timestamps of the recorded spikes so that the delay through the filters is removed.

The 2D feature matrix of a speech sample within a time window (Fig. 4) shows the counts for all channel neurons in an 80-bin inter-spike interval (ISI) histogram. Channel responses of both ears are used even though this is not necessary for our speech task. Since the directly resulting feature vector has a high dimensionality (80 (the number

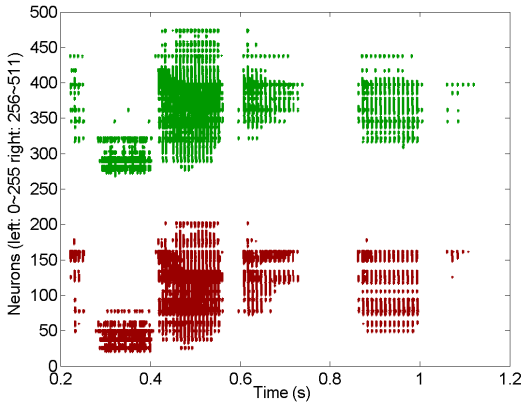

Fig. 3. Response to speech "The shadow vanished". Colors correspond to channels of the different ears. Each dot is one event.

of bins) times 512 (the number of neurons)), a new 1D feature vector consisting of two concatenated 1D vectors is extracted from this 2D matrix. The first part, shown below the matrix, is the ISI distribution collapsed across channels using 80 bins. The second part, shown to the right of the matrix, is the average activity of the individual channels.

Furthermore, a single output is generated for each channel by taking the average activity of the 4 neurons so that the final dimension is reduced to 80 plus the number of channels. This averaging step also reduces the noise in the spike times.

IV. SPEAKER IDENTIFICATION

Feature vectors are extracted for a particular time window. During the training phase, features vectors generated from the training samples are fed into a LIBSVM support vector machine (**SVM**) for classification [12]. During the test phase, feature vectors from the test samples are fed into the SVM which then output the probabilities of the learned speakers for each feature vector extracted in a time window. The most probable speaker is determined at the end of the sentence by summing the probabilities computed over all feature

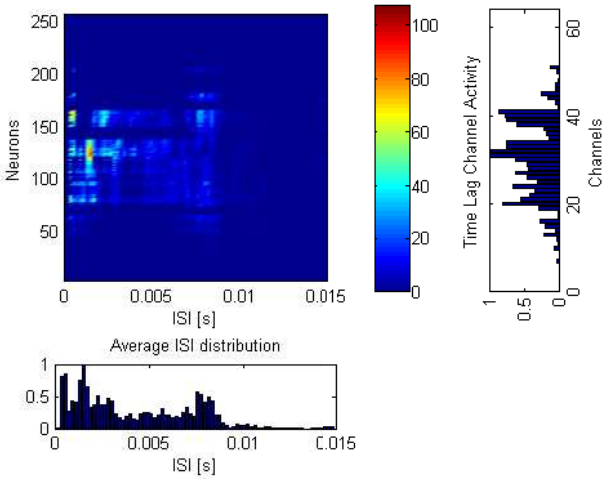

Fig. 4. Inter-spike intervals and feature histograms for a male speaker, from combined neurons of left and right cochlea channels.

vectors at the end of the sentence. The training and testing procedure is repeated 10 times, by excluding one sentence from each speaker from the training samples, and testing on the excluded sentence.

In the experiments described below, 40 speakers (20 males and 20 females) and 10 sentences per speaker are selected from the TIMIT. The length of the sentences varies between 0.8 to 4 seconds.

In this study, two methods that can reduce the computation for feature extraction during real-time speaker identification are considered. In Method 1, feature vectors are computed for every 100ms time window only if the spikes in the window exceed a preset threshold **Th1.** In Method 2, a feature vector is computed only when the number of spikes exceeds a threshold **Th2** irrespective of the length of the time window. A fading time constant $\boldsymbol{\tau}$ is also applied to the computed histograms.

V. RESULTS

As shown in Table 1, the performance shows a 85%-96% correct identification of the correct speaker from the 40 subjects for all methods. These numbers vary depending on the subset of 40 speakers chosen from the complete 400 speakers in the database. The results for Method 1 show that time averaging for a fading histogram decreases the classification performance, and that the classification performance for Method 2 and a **Th2** of 800 spikes does not change significantly even though 50% fewer vectors were used. These results suggest that we can perform our classification without computing feature vectors at regular time windows and that performance is better if we do not low-pass filter the ISI histogram feature vector. Interestingly, the inclusion of second-order ISIs histogram vectors (ISI order) does not affect the performance significantly.

Table 1: Speaker identification performance, in percent correct of 40 speakers.

		τ (ms)					
Window	ISI order and Th1/2	0	400	800	1200	1600	2000
Fixed window length 100ms	1st, Th1=0	98.50	92.50	91.50	91.25	91.00	90.50
	1st & 2nd, Th1=0	98.25	92.75	91.75	90.50	90.50	90.25
	1st, Th1=20	95.75	86.50	86.00	85.50	85.50	85.00
	1st & 2nd, Th1=20	95.50	88.25	87.00	85.50	85.75	85.50
Variable Window Length	1st, Th2=800	95.50	92.50	91.50	90.25	89.50	88.75
	1st & 2nd, Th2=800	95.75	91.50	89.50	89.50	89.00	89.00

While the results in Table 1 are computed after the end of a test sentence; in a normal scenario, it is difficult to determine when the speaker has finished a sentence. Hence the performance of the system is analyzed in a more natural scenario by concatenating sentences from the 40 speakers and determining how quickly the correct classification is determined during presentation. It is found that a "cumulative" probability measure that is computed as follows results in rapid and reliable speaker

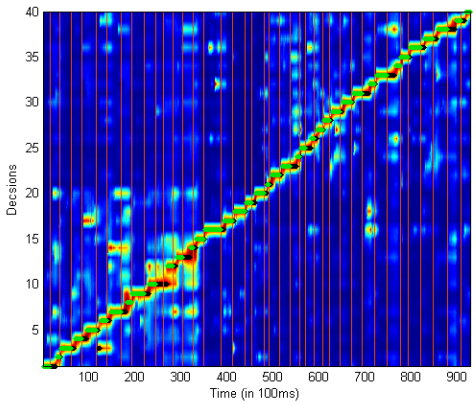

Fig. 5. Cumulative measure of the 40 speakers (y-axis) determined over 40 concatenated sentence files from the 40 speakers. Vertical red lines demarcate the change from one speaker to another.

classification:

$$P(t) = p(t) \cdot max\,(p(t)) + P(t-1)e^{(\frac{-\Delta t}{\tau_p})}$$

where $max\,(p(t))$ is the maximum of the 40 instantaneous probabilities $p(t)$ in a current window and τ_p determines how much the past cumulative probabilities $P(t-1)$ determine the final probabilities $P(t)$ at a time step. These probabilities in turn determine the run-time classifications. The term $max\,(p(t))$ weighs the update of cumulative values so that if the current probability is high, it will have a larger effect on the running probabilities. Thus highly-confident classification results are weighted more strongly.

Fig. 5 shows a plot of the running cumulative measure of the 40 speakers over randomly chosen sentences from each of the 40 speakers and Fig. 6 shows a close-up of one transition. Fig. 7 shows a close-up of a transition when Method 2 is used. Here the decisions are taken when feature vectors are extracted, and not at regular intervals as in Fig. 6.

Because the training and testing procedures are each repeated 10 times, by excluding one sentence from each speaker from the training samples, and testing on the excluded sentence, there are 10 sets of data for the cumulative probability analysis. On average, the run-time decision accuracy as seen at the end of each sentence is 92%. The cumulative probability plot shown in Fig. 5 is from the best set with 97.5% accuracy.

The latency for the classifier to make the correct

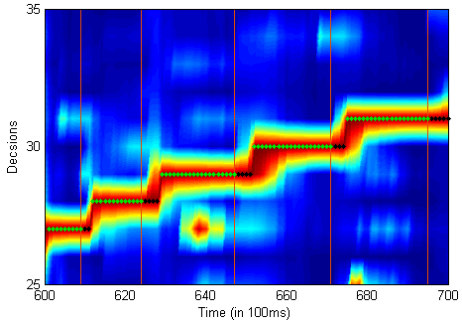

Fig. 6. Close-up of a section in Fig. 5. Black dots denote the wrong decisions. Green dots denote the correct decisions.

decision is also measured when the algorithm encounters a new test file from a new speaker. This latency is similar to what the algorithm would face in a normal speech condition when multiple speakers are present in a conversation. The latency from 10 run-time test sessions is 720±150ms when τ_p =0.3s.

In both methods, the time constant τ_p, which determines how much the cumulative probabilities should depend on the past values, affects both the latency and the accuracy. When the time constant is small, the latency to the correct decision is small but the algorithm also makes more incorrect decisions as shown in Fig. 8.

But on the other extreme, when τ_p is very large, the performance remained around 90% with a latency of ~900ms. When τ_p=10s, average latency is 900ms, and accuracy is 89.75% (in 100ms fixed window case).

The second method with the variable length window requires a larger time constant because it generally leads to fewer segments per sentence than the 100ms window method, or in other words, a longer average window length, as can be inferred from the following comparison.

The algorithm generated 9233 feature vectors from the 400 sentences using the first method of a fixed 100ms window (only feature vectors with non-zero spikes are kept), 5429 feature vectors if we keep only the feature vectors where there were more than 20 spikes in the 100ms windows, and 4951 vectors using the second method and with **Th2**= 800 spikes.

VI. CONCLUSION

The availability of novel spike-based cochlea systems with integrated functionality and usability allows us to investigate real-time auditory processing (for e.g. spatial audition and speech recognition) in a way similar to that of biological systems. This work shows the performance of a system using features suitable for run-time performance in the speaker identification task. We find that the time constant of the cumulative probabilities determines the tradeoff between the accuracy and latency of the classification in a test condition approximating normal speech conditions. Our latency numbers are based on the use of a 100ms time window so that we can speed up the computation. The tradeoff between the performance of smaller time windows and the latency to correct decision will be explored in the future. Implementation of the algorithm in the project-based JAVA environment will allow us to test these features in a real-time speech scenario.

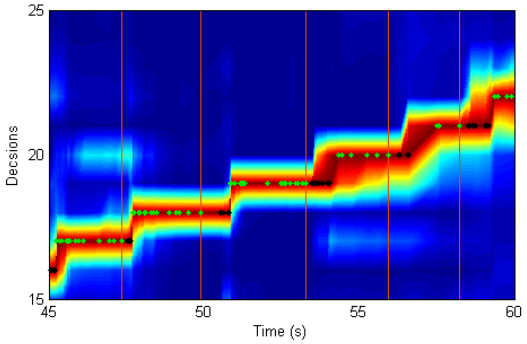

Fig. 7. Close-up of a section of the plot when the second method is used. The green dots do not look as dense as is Fig. 6 because the feature vectors are not generated at fixed intervals. τ_p is 700ms.

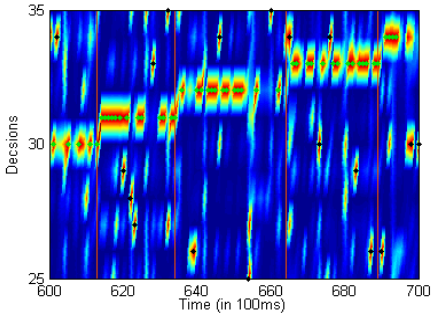

Fig. 8. Results of first method when τ_p is 50ms.

ACKNOWLEDGEMENT

The authors would like to thank Mert Yentur for initial experiments with these feature vectors in jAER. This project was partially funded by the Swiss National Science Foundation grant #200021-126844 "Early Auditory Based Recognition of Speech".

REFERENCES

[1] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. V. Campenhout, "Isolated word recognition with the liquid state machine: a case study," Information Processing Letters, vol. 95, pp. 521–528, 2005.

[2] I. Uysal, H. Sathyendra, and J. G. Harris, "A biologically plausible system approach for noise robust vowel recognition," *IEEE Proc. of the Midwest Symposium on Circuits and Systems*, vol. 1, pp.245-249, 2006.

[3] B. Wen and K. Boahen, "A 360-channel speech preprocessor that emulates the cochlear amplifier," in *ISSCC Dig. of Tech. Papers*, 2006, pp. 556-557.

[4] R. Sarpeshkar, et al., "An analog bionic ear processor with zero-crossing detection," in *ISSCC Dig. of Tech. Papers*, 2005, pp. 78-79.

[5] E. Fragniere, "A 100-channel analog CMOS auditory filter bank for speech recognition," in *ISSCC Dig. of Tech. Papers,* 2005, pp. 140-589.

[6] N. Kumar, W. Himmelbauer, G. Cauwenberghs, and A. G. Andreou, "An analog VLSI chip with asynchronous interface for auditory feature extraction," *IEEE Transactions On Circuits And Systems—II: Analog and Digital Signal Processing*, vol. 45, no. 5, May 1998, pp. 600-606.

[7] H. Abdalla and T. Horiuchi, "An ultrasonic filterbank with spiking neurons", *Proceedings of IEEE International Circuits and Systems 2005,* vol. 5, 2005, pp. 4201–4204.

[8] S.-C. Liu, A. van Schaik, B. Minch, and T. Delbruck, "Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms," *Proceedings of IEEE International Circuits and Systems,* 2010, pp. 2027-2030.

[9] Welcome to the jAER Open Source Project. Retrieved 10.10.2011. Available: jaer.wiki.sourceforge.net.

[10] S.-C. Liu, N. Mesgarani, J. Harris, and H. Hermansky, "The use of spike-based representation for hardware audition systems", *Proceedings of IEEE International Circuits and Systems,* 2010, pp. 505-508.

[11] S. Chakrabartty and S.-C. Liu, "Exploiting spike-based dynamics in a silicon cochlea for speaker identification," *Proceedings of IEEE International Circuits and Systems 2010,* pp. 513-516.

[12] C-C Chang and C-J Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.